Research monograph

Linear Models for Nonlinear Dynamics

Victor M. Preciado
University of Pennsylvania

Foundations, representations, and finite models

Free Sample

Review draft: 19 November 2026

Complete draft: all 23 chapters, with executable labs and selected worked solutions.

How to read this book

How can a finite linear model help us understand nonlinear motion? First choose functions or distributions in which evolution is linear. Then choose the finite information to keep and account for what it leaves out. This is the organizing idea of *lift, then project*.

You need calculus, basic matrix algebra, and introductory ordinary differential equations. Functional analysis, stochastic calculus, operator methods, and model reduction are introduced through worked examples. The identification route also uses least squares and numerical array operations developed in Chapters 4 and 10. Compression in Chapter 13 asks you to work with transfer functions and Lyapunov equations; pause at its worked examples if those are new. The control route introduces convex quadratic optimization in Chapter 20. Technical operator and stochastic proofs have separate reading paths. Advanced results state their assumptions and identify the sources of their proofs.

Chapter 1 derives a finite lift line by line. Chapters 2–8 build the matrix, approximation, probability, and transport foundations. Eliminating a hidden variable in Chapter 9 then explains memory. Chapters 10–14 add fitting, delays, memory models, compression, and bounds. Chapters 15–19 develop operator background, exactness and spectra, diffusions, distributed systems, and controlled lifts. The final chapters develop predictive control, observers, integrated case studies, and task-based model selection. Read the worked examples before the abstract qualifications.

Figures support the derivations. Schematics explain dependencies; plots come from checked calculations with disclosed parameters. The stochastic lab reports sampling error separately. Teaching examples illustrate mechanisms and do not establish generic empirical superiority.

Every chapter includes exercises, an executable Python lab, and selected solutions. Exercises identify their intended reading route. Additional labs develop

Free Sample

the mathematical bridges, noisy pendulum identification, and a spatial-resolution study. Reading notes attribute classical constructions; the book claims no new lifting theorem or reduction algorithm. The complete source bundle includes all chapter sources, figures, data, and executable checks.

Free Sample

Reading routes and notation

Choose a route, and know when to pause

The common entry route is Chapters 1–6: construct a lift, review matrix tools, select coordinates, project, evolve observables, and examine a polynomial hierarchy that can fail. The three-measurement fit in Section 4.5.1 and the complete evolution-matrix fit in Section 10.3 bridge scalar regression and prediction. You should be able to initialize a dictionary, multiply a value vector, and explain a weighted residual before continuing.

For a prediction and identification course, continue through the finite memory calculation in Chapter 9 and Chapters 10–12, then the noisy engineering case in Section 22.7. Read Chapters 7–8 before interpreting probability laws or population projections. Chapters 13–14 then introduce compression and bounding as different output tasks.

For a control route, follow the common entry and fitting chapters with Chapters 19–21. Return to Chapter 13 for transfer reduction and reset qualifications. The engineering case studies and Chapter 23 connect these choices. The operator background is useful for understanding qualifications; its technical proofs are not prerequisites for the finite control calculations.

For the stochastic and distributed route, complete Chapters 7–9, then work through the Fourier bridge in Section 15.2. Chapters 15–18 have an engineering path and a technical path. First calculate with profiles, modes, Gaussian increments, and finite spatial cuts. Then return to generator-domain and general existence arguments. The stochastic-integral bridge in Section 17.2 constructs the deterministic-weight integral needed for OU before stating the full chain rule. Imported theorems identify their proof sources.

Free Sample

Exercise and laboratory conventions

Exercises are marked **Practice** when they apply a worked method, **Connect** when they combine concepts or justify assumptions, and **Explore** when they extend the example or design a new test. **Technical** marks selected advanced operator arguments that can be deferred. These are reading routes, not time estimates or grades. Selected solutions demonstrate reasoning; an unsolved exercise is not an unstated prerequisite for the next chapter.

Before each lab, write a prediction. Implement one small calculation, run the supplied reference, and explain the difference. The numerical practice appendix, Appendix A, gives setup, interpretation, and troubleshooting guidance. The annotated notebook in Appendix A.2 reviews the needed Python array operations before fitting a model. The code includes explicit checks, but passing checks is not the whole learning task. The companion self-study guide in Appendix A.2.7 maps errors and assessment tasks to recovery steps. A new transfer record tests whether the snapshot construction can be generalized beyond the worked example. A second problem in Appendix A.2.6 changes the system, coordinate dictionary, and decoder, with a separate solution for feedback. For readers following the common route, Equation (6.3) already solves this system, and Equation (16.1) supplies its coordinate evolution. Treat the task as implementation and interpretation transfer. A fresh derivation attempt must precede both passages and the key; record prior exposure separately from hints. No classroom reader pilot has yet been completed; these routes are an instructional design to be tested with students.

Free Sample

A compact symbol glossary

Symbols have the following default meanings. A local example may reuse a scalar letter after defining it; the roles of operators, matrices, projections, and coordinates remain distinct.

Symbol	Meaning and first useful destination
$x, \widehat{x}$	Physical state and predicted state; Chapter 1.
φ^t	State flow for duration t, on its stated domain.
ψ, V_N	Ordered dictionary of functions and its linear span.

Symbol	Meaning and first useful destination
$z = \psi(x)$	Values of dictionary functions, initialized from a physical state.
C	Linear readout of the requested output from feature values.
A	Finite coordinate-value generator: $\dot{z} = Az$.
M	Sampled coordinate-value matrix: $z^+ = Mz$.
K_N	Row-data regression matrix, with value matrix $M = K_N^\top$.
$\mathcal{K}, \mathcal{K}^t$	Observable evolution operators; Chapter 5.
$\mathcal{L}, \operatorname{dom}\mathcal{L}$	Observable generator and the functions on which its norm-limit derivative exists.
$\mathcal{L}^*$	Adjoint with its stated pairing: distribution duality in Chapter 8; weighted Hilbert inner product in Chapter 15. Boundary and domain conditions belong to that choice.
$\mathcal{B}_*$	Basin of the specified attracting equilibrium; Theorem 16.2.
$\chi(t)$	Geometric finite-cut ratio $x_0(1 - e^{-t})$; Chapters 6 and 22.
$r(x)$	Rational scalar coordinate $x/(1 - x)$; Chapter 16 and the second transfer task.
φ_j	Normalized Hermite mode; distinct from sample interval h.
H_U	Horizon objective Hessian; distinct from value matrix M.
n_{pair}	Number of training snapshot pairs; distinct from mechanical mass m.
$E_{\text{train,phys}}$	Normalized physical-state one-step training RMS, distinct from the fitting loss.
P, Q	Projection onto retained information and its complement $I - P$.
D	Samples-by-features design matrix.
G_ψ	Weighted dictionary Gram matrix; Chapter 4.
λ_{R}	Nonnegative regularization strength, with penalty convention stated locally.
$\|\cdot\|, \|\cdot\|_2$	Specified norm; Euclidean norm for finite vectors and integral-square norm for functions with their stated measure.
$\langle g, h\rangle$	Inner product, linear in the first argument here; complex conjugation acts on the second.
$\mathbb{1}_B$	Indicator, equal to one on B and zero elsewhere.
$\mathbb{E}, \operatorname{Var}$	Expectation and variance under a stated probability law.
ρ, μ	Density and measure; measures need not possess densities.

Free Sample

Symbol	Meaning and first useful destination
T, G	Row-stochastic transition matrix and continuous-time rate matrix. Probability columns evolve with their transposes.
y, w	Retained and hidden states in the finite elimination example.
k	Memory kernel for a stated projection; Chapter 9.
$\mathcal{L}_\perp$	Generator of the specified unresolved evolution, if it exists.
W_c, W_o	Controllability and observability Gramians.
η_j	Ordered Hankel singular values, distinct from eigenvalues.
Π_m	Spatial state projection, distinct from observable projection P.
Γ, Θ	Output-injection gain and driven-filter reconstruction map.

Recurring systems and parameter choices

Name	Equation, convention, and purpose
S1, autonomous	$\dot{x}_1 = \kappa_1 x_1$, $\dot{x}_2 = \kappa_2(x_2 - x_1^2)$. Teaching rates are $(-1, -3)$; closed dictionary (x_1, x_2, x_1^2).
S1, controlled	Add u to the first equation. Teaching rates remain $(-1, -3)$ in Chapters 13, 19 and the analytic examples of Chapters 20 and 23.
S1, research grid	In frozen code, `mu`$= \kappa_1$ and `lambda`$= \kappa_2$. Rates $(-0.05, -1)$ in the nine-case MPC comparison. Equilibrium input is $u_* = -\kappa_1 a$. The teaching-rate formulas must be rescaled before reuse.
S2	$\dot{x} = -x + x^2$, used for polynomial-cut convergence and nonlinear reconstruction. Its existence interval depends on initialization.
Finite memory pair	$\dot{y} = -y + w$, $\dot{w} = y - 2w$ in Chapter 9.
Two-mode correlation	$c = (e^{-t} + e^{-2t})/2$, from rotated deterministic modes in Chapter 12 and stationary OU observables in Chapter 17.
Logistic obstruction	$\dot{x} = x(1 - x)$ on $[0, 1]$; Section 16.5. Its sign is the reverse of S2.
Noisy pendulum	$\dot{\theta} = \omega$, $\dot{\omega} = -0.2\omega - \sin\theta$; two noisy sensors and nonclosed fitted features in Section 22.7.

Free Sample

Name	Equation, convention, and purpose
Burgers resolution	Periodic $u_t = 0.2u_{xx} - uu_x$, initial $0.8\sin x + 0.3\sin 2x$; Section 18.6.

Free Sample

Contents

How to read this book i

Reading routes and notation iii

1 Why finite linear models? 1
- 1.1 A forcing that a tangent model misses . . . 1
- 1.2 Add the missing coordinate . . . 3
- 1.3 Initialization, evolution, and readout are separate steps . . . 5
- 1.4 Why functions can evolve linearly . . . 7
- 1.5 What happens when closure fails? . . . 9
- 1.6 A workflow for the rest of the book . . . 10
- 1.7 Guided lab: three routes to the same trajectory . . . 11
- 1.8 Exercises . . . 12
- 1.9 Notes and further reading . . . 13

2 Linear dynamics and the matrix tools we need 14
- 2.1 A two-body thermal calculation . . . 14
- 2.2 Matrix exponentials and forcing . . . 15
- 2.3 Modes and changes of coordinates . . . 17
- 2.4 Repeated rates and Jordan blocks . . . 18
- 2.5 Norms, singular values, and conditioning . . . 19
- 2.6 Stable eigenvalues can coexist with transient growth . . . 20
- 2.7 Guided lab: solve, change coordinates, and measure amplification . . . 21
- 2.8 Exercises . . . 22
- 2.9 Notes and further reading . . . 23

3 Functions as coordinates 24
- 3.1 A dictionary is a bank of functions . . . 24
- 3.2 Independence belongs to a domain and to the data . . . 25
- 3.3 Constants turn affine equations into homogeneous linear equations . . . 27
- 3.4 Can the coordinates recover the requested output? . . . 28
- 3.5 Linear evolution with a nonlinear decoder . . . 28
- 3.6 A coordinate specification that can be audited . . . 30
- 3.7 Guided lab: feature rank, offsets, and reconstruction . . . 30
- 3.8 Exercises . . . 31

Free Sample

3.9 Notes and further reading . . . 32

4 Approximation, projection, and least squares 33
4.1 The closest point on a line . . . 33
4.2 Fit several functions to sampled values . . . 34
4.3 From weighted samples to a distribution of states . . . 35
4.4 Worked example: a quadratic approximated by a line . . . 36
4.5 Solve the geometry without unnecessarily squaring the conditioning . . . 37
4.6 A training residual is weighted evidence . . . 40
4.7 Guided lab: weighted projection and an independent check . . . 41
4.8 Exercises . . . 42
4.9 Notes and further reading . . . 43

5 Evolution of observables 44
5.1 One step: substitute the next state into a function . . . 44
5.2 The generator is the chain rule written as a map . . . 45
5.3 Eigenfunctions are measurements with simple time dependence . . . 46
5.4 When does a finite dictionary evolve exactly? . . . 48
5.5 Value matrices and coefficient matrices are transposes . . . 49
5.6 A periodic example without polynomial coordinates . . . 49
5.7 What is deferred to the operator-theory chapters? . . . 50
5.8 Guided lab: verify both orientations . . . 51
5.9 Exercises . . . 51
5.10 Notes and further reading . . . 52

6 Polynomial lifts and Carleman hierarchies 53
6.1 A quadratic equation generates a ladder of powers . . . 53
6.2 Construct the finite matrix before discussing its accuracy . . . 54
6.3 Solve the nonlinear equation and the finite cut . . . 55
6.4 Convergence has a state region and a horizon . . . 57
6.5 Degree truncation and least-squares closure are different . . . 58
6.6 Multivariate monomials and model size . . . 58
6.7 Guided lab: check a remainder independently of its derivation . . . 59
6.8 Exercises . . . 60
6.9 Notes and further reading . . . 61

7 Probability and finite Markov chains 62
7.1 Random states and weighted averages . . . 62
7.2 Discrete-time transition matrices . . . 64
7.3 Continuous-time rates and probability balance . . . 64
7.4 Enumerate a two-node epidemic exactly . . . 65
7.5 Moments expose the closure choice . . . 67
7.6 A projection can lose the Markov property . . . 68
7.7 Guided lab: law, moments, and closure . . . 69
7.8 Exercises . . . 69
7.9 Notes and further reading . . . 70

Free Sample

8 Distributions, transport, and duality **71**
8.1 From particles to probabilities . . . 71
8.2 A contracting ensemble has a growing density . . . 72
8.3 Derive the continuity equation by balancing flux . . . 73
8.4 Weak balance makes the boundary assumptions explicit . . . 74
8.5 Duality: move measurements or move probabilities . . . 75
8.6 Point masses and empirical distributions . . . 76
8.7 Open boundaries change the mass budget . . . 77
8.8 Guided lab: verify transport by two independent integrations . . . 78
8.9 Exercises . . . 78
8.10 Notes and further reading . . . 79

9 Projection, residuals, and memory **80**
9.1 Bridge from the foundations . . . 80
9.2 Two states, one retained variable . . . 81
9.3 Eliminate the hidden state exactly . . . 83
9.4 The block-matrix identity . . . 85
9.5 Where projection and the residual enter . . . 87
9.6 Discard, model, or minimize the unresolved effects . . . 88
9.7 From residual forcing to output error . . . 90
9.8 What this calculation establishes . . . 92
9.9 Guided lab: eliminate, realize, and check . . . 92
9.10 Exercises . . . 93
9.11 Notes and further reading . . . 94

10 Learning finite evolution matrices **96**
10.1 Choose the prediction task before assembling data . . . 96
10.2 Derive the least-squares operator . . . 97
10.3 Work one complete evolution fit by hand . . . 98
10.4 An exact fitted lift and a finite physical baseline . . . 102
10.5 One-step residuals do not propagate themselves . . . 103
10.6 Generator regression is a different experiment . . . 104
10.7 A fit can be exact on the wrong support . . . 105
10.8 Noise, scaling, and an honest fitting workflow . . . 105
10.9 A bridge to measured engineering data . . . 106
10.10 Guided lab: fit and audit both matrices . . . 107
10.11 Exercises . . . 107
10.12 Notes and further reading . . . 108

11 Better coordinates and delay coordinates **109**
11.1 Past measurements can distinguish present states . . . 109
11.2 An exact recurrence for one harmonic . . . 110
11.3 Delay regression is another projection problem . . . 111
11.4 Two harmonics require more information . . . 112
11.5 An exact delay model is not a truncated Mori kernel . . . 114

Free Sample

11.6 Changing features can reduce a residual and change the task 114
11.7 Guided lab: forecast and interpret 115
11.8 Exercises 116
11.9 Notes and further reading 117

12 Modeling memory **118**
12.1 Start from a memory equation with a declared projection 118
12.2 An exponential kernel is a linear filter 119
12.3 Derive a two-mode signal before fitting anything 120
12.4 Fit only the declared training interval 120
12.5 Parameters, kernels, and trajectories are different checks 122
12.6 A sufficient bound for propagating kernel error 123
12.7 Guided lab: fit a family, then test its consequences 123
12.8 Exercises 125
12.9 Notes and further reading 125

13 Compressing a finite input-output model **127**
13.1 An input-output primer 127
13.2 Reachability and observability are task properties 129
13.3 A lifted comparator with an unreachable reset coordinate 129
13.4 Physical initialization can restore the deleted direction 130
13.5 Gramians measure input access and output visibility 131
13.6 Balanced coordinates and one approximate reduction 132
13.7 Guided lab: transfer, initialization, and balancing 136
13.8 Exercises 136
13.9 Notes and further reading 137

14 Bounding without reconstructing a trajectory **138**
14.1 Ask for an accumulated quantity 138
14.2 Occupation measures record time spent in sets 139
14.3 Turn inequalities into an upper certificate 140
14.4 A linear program on measures and its relaxations 143
14.5 A small moment problem that can be solved by hand 143
14.6 Polynomial positivity and numerical tolerances 144
14.7 Guided lab: verify a bound without fitting a predictor 146
14.8 Exercises 146
14.9 Notes and further reading 147

15 From matrices to operators **148**
15.1 A function space includes its norm 148
15.2 Bridge: from finite vectors to a temperature profile 149
15.3 Moving a profile without differentiating it 152
15.4 The generator needs a domain 153
15.5 Adjoints expose boundary conditions 155
15.6 Technical path: when the operator memory formula is justified 156
15.7 Guided lab: check the finite version independently 158

Free Sample

15.8 Exercises . . . 159
15.9 Notes and further reading . . . 160

16 Exactness, spectra, and resonance 161
16.1 Three questions hidden inside “exact” . . . 161
16.2 Resonance preserves closure but can destroy diagonalization . . . 162
16.3 Changing the function space changes the eigenfunctions . . . 164
16.4 A finite eigenvalue can be an artifact of truncation . . . 164
16.5 A worked obstruction: two endpoints of one bounded orbit . . . 166
16.6 Technical path: smooth linearization guarantees . . . 168
16.7 Guided lab: inspect coordinates and residuals . . . 170
16.8 Exercises . . . 171
16.9 Notes and further reading . . . 172

17 Stochastic dynamics and diffusions 173
17.1 Why random increments scale as the square root of time . . . 174
17.2 Bridge: build a noise integral from increments . . . 174
17.3 The stochastic chain rule . . . 177
17.4 Work out noisy relaxation completely . . . 178
17.5 Forward evolution and equilibrium . . . 179
17.6 Hermite modes explain the memory precursor . . . 180
17.7 Finite samples are a separate source of error . . . 181
17.8 Guided lab: population checks before sampling checks . . . 182
17.9 Exercises . . . 183
17.10Notes and further reading . . . 184

18 Distributed systems and two projections 185
18.1 A field evolves by local balance . . . 185
18.2 Spatial modes turn the field into coordinates . . . 186
18.3 The time integrator can introduce its own instability . . . 187
18.4 A nonlinear spatial model still needs an observable lift . . . 187
18.5 Add errors only after putting them in the same task . . . 189
18.6 Construct a spatial-resolution study . . . 191
18.7 Extensions: advection and nonlinear reconstruction . . . 193
18.8 Guided lab: keep the three references distinct . . . 193
18.9 Exercises . . . 194
18.10Notes and further reading . . . 194

19 Inputs and bilinear lifts 195
19.1 Inputs change the closure test . . . 195
19.2 Derive the exact controlled two-state lift . . . 196
19.3 A held input permits exact propagation . . . 197
19.4 Two affine comparators retain different information . . . 198
19.5 A nonlinear reset distinguishes the affine models . . . 199
19.6 Guided lab: compare prescribed-input predictions fairly . . . 200
19.7 Exercises . . . 201

Free Sample

19.8 Notes and further reading . . . 202

20 Predictive control and robustness 203
20.1 Plan several moves, apply one, and measure again . . . 203
20.2 Work out a convex problem before a nonlinear one . . . 204
20.3 A shared output task for three controlled predictors . . . 206
20.4 Separate start sensitivity from stopping sensitivity . . . 207
20.5 Prove a robustness margin on a specified region . . . 209
20.6 Guided lab: optimization checks and a separate certificate . . . 210
20.7 Exercises . . . 210
20.8 Notes and further reading . . . 211

21 Observers and state reconstruction 212
21.1 A measured coordinate can reveal an unmeasured one . . . 212
21.2 Choose poles, then inspect transients and noise . . . 213
21.3 A closed measurement need not recover the state . . . 214
21.4 Include model residuals in the observer error . . . 215
21.5 Stable filters can have an exact reconstruction map . . . 216
21.6 What the nonlinear construction must assume . . . 217
21.7 Guided lab: check the error equation before tuning the gain . . . 218
21.8 Exercises . . . 218
21.9 Notes and further reading . . . 219

22 Integrated computational case studies 220
22.1 Begin with the claim and its units . . . 220
22.2 Case one: does increasing a lift order help this trajectory? . . . 221
22.3 Case two: can the training support hide missing dynamics? . . . 222
22.4 Cases three through five: match evidence to the task . . . 223
22.5 Case six: audit the physical control cost from raw records . . . 224
22.6 Guided lab: rerun and audit the whole evidence chain . . . 226
22.7 A sustained engineering case: learning a noisy pendulum . . . 226
22.8 What a broader benchmark would still need . . . 238
22.9 Exercises . . . 239
22.10 Notes and further reading . . . 239

23 Choosing a modeling strategy 240
23.1 Describe the task before selecting the coordinates . . . 240
23.2 Decision one: how much hidden-variable memory is needed? . . . 241
23.3 Decision two: preserve a convex input-prediction problem . . . 242
23.4 Decision three: certify exposure instead of predicting a path . . . 244
23.5 Put the rest of the error budget back in . . . 245
23.6 Guided lab: record choices and the reasons behind them . . . 246
23.7 A fourth decision: when data choose the feature budget . . . 246
23.8 Questions to carry into a new application . . . 247
23.9 Exercises . . . 247
23.10 Notes and further reading . . . 248

Free Sample

A Laboratory practice and numerical accuracy **249**
A.1 Set up once, then run one small example 249
A.2 An annotated notebook from arrays to a forecast 251
A.3 Read a lab in four passes 257
A.4 What the tolerances actually control 258
A.5 Reference, sampling, and model errors 259
A.6 Troubleshoot before retuning 259
A.7 A proof-status map 260

Selected solutions **261**

Index **281**

Free Sample

Chapter 1

Why finite linear models?

Suppose a sensor records a quantity driven by the square of a second quantity. A linearization at equilibrium may remove that forcing entirely. Could a linear model still retain it? Yes, if its coordinates include the square itself. This simple observation is our starting point.

The aim is not to turn every nonlinear problem into a small matrix problem. It is to learn how to recognize the useful linear descriptions, construct finite models from them, and state what those models leave unresolved. We will repeatedly ask which output, initial conditions, and time horizon a proposed model is meant to handle.

What you will be able to do

Differentiate a proposed nonlinear coordinate; verify an exact lifted matrix equation; initialize and read out a lifted state; distinguish a lift from a tangent linearization; and explain why linear evolution of functions does not guarantee a finite closed model.

Free Sample

1.1 A forcing that a tangent model misses

We call the following two-state teaching system S1. Consider its ordinary differential equation

$$\dot{x}_1 = \kappa_1 x_1, \qquad \dot{x}_2 = \kappa_2(x_2 - x_1^2), \qquad \kappa_1, \kappa_2 < 0. \tag{1.1}$$

A dot means differentiation with respect to time. The constants κ_1 and κ_2 are rates, with units of inverse time after the states have been nondimensionalized. The first

state decays independently. The second tends to follow the moving target x_1^2. Indeed, because $\kappa_2 < 0$, its derivative has the sign of $x_1^2 - x_2$. This is a simple cascade, rather than a detailed model of a particular device. It captures a common engineering situation: a decaying source drives a second process through a nonlinear intensity or reaction term.

Use the teaching parameters

$$\kappa_1 = -1, \qquad \kappa_2 = -3, \qquad x_1(0) = 1, \quad x_2(0) = 0. \tag{1.2}$$

Time is measured in the corresponding reference time unit. The first equation gives $x_1(t) = e^{-t}$, so the second becomes $\dot{x}_2 + 3x_2 = 3e^{-2t}$. Multiplying by e^{3t} gives

$$\frac{\mathrm{d}}{\mathrm{d}t}\big(e^{3t}x_2(t)\big) = 3e^{t}.$$

Integrate from zero to t, apply $x_2(0) = 0$, and divide by e^{3t}:

$$x_1(t) = e^{-t}, \qquad x_2(t) = 3(e^{-2t} - e^{-3t}). \tag{1.3}$$

Although the eventual equilibrium is zero, x_2 first rises. Its peak occurs when $-2e^{-2t} + 3e^{-3t} = 0$, or $t = \log(3/2)$, and is $4/9$. Both the transient rise and its timing can matter to a design.

A first-order Taylor approximation at $(0, 0)$ uses the Jacobian matrix, the matrix of partial derivatives of the vector field:

$$Df(0) = \begin{pmatrix} -1 & 0 \\ 0 & -3 \end{pmatrix}.$$

Writing its prediction as $\widehat{x}$, the tangent model is $\dot{\widehat{x}}_1 = -\widehat{x}_1$ and $\dot{\widehat{x}}_2 = -3\widehat{x}_2$. Initialized at $(1, 0)$, it predicts $\widehat{x}_2(t) = 0$ for every t. The quadratic source has disappeared. Figure 1.1 shows the consequence. A first-order Taylor approximation concerns small deviations near its expansion point. The chosen initial deviation is not small; the calculation deliberately exposes what the approximation removes.

Free Sample

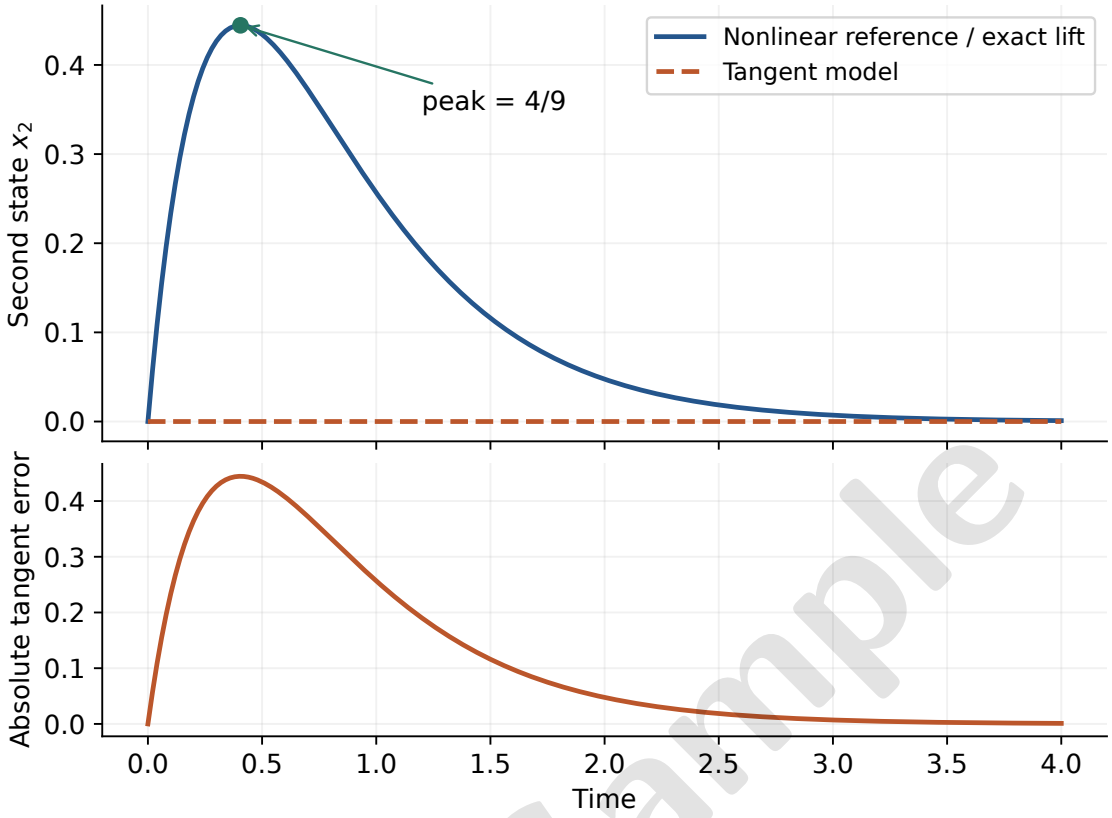


Figure 1.1: A nonlinear forcing can create a transient absent from the tangent model. For Equation (1.2), the exact lifted output and the nonlinear reference coincide. The tangent model predicts zero for x_2. The lower panel isolates the resulting absolute output error. This is an analytic teaching example, not a learned-model benchmark.

1.2 Add the missing coordinate

The source x_1^2 suggests an additional coordinate. Define the *lift*, a map from physical states into a larger coordinate vector,

$$z = \psi(x) = \begin{pmatrix} x_1 \\ x_2 \\ x_1^2 \end{pmatrix}. \tag{1.4}$$

Free Sample

The first two entries are unchanged. The third is a function of the first, not a new physical degree of freedom. Differentiate each entry:

$$\dot{z}_1 = \dot{x}_1 = \kappa_1 z_1,$$
$$\dot{z}_2 = \dot{x}_2 = \kappa_2 z_2 - \kappa_2 z_3,$$
$$\dot{z}_3 = 2x_1\dot{x}_1 = 2\kappa_1 x_1^2 = 2\kappa_1 z_3.$$

Each derivative is a linear combination of the same three entries. Therefore the lifted coordinates satisfy the constant matrix equation

$$\dot{z} = Az, \qquad A = \begin{pmatrix} \kappa_1 & 0 & 0 \\ 0 & \kappa_2 & -\kappa_2 \\ 0 & 0 & 2\kappa_1 \end{pmatrix}. \tag{1.5}$$

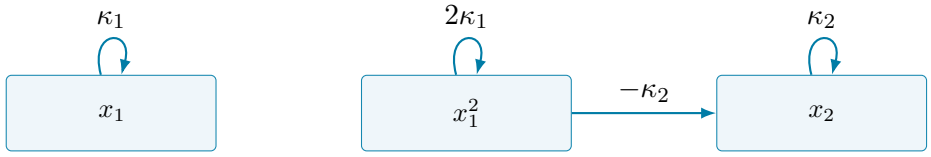
This derivation used the chain rule, not a Taylor approximation. The equation is exact for every initial state of Equation (1.1), with the initialization rule in Equation (1.4).

A vector of functions is a dictionary

The functions $\psi_1(x) = x_1$, $\psi_2(x) = x_2$, and $\psi_3(x) = x_1^2$ are called *observables*: quantities that can be evaluated at a state. Their ordered collection is a *dictionary*. The span of a dictionary is the set of all constant linear combinations of its functions. Here it consists of $a_1x_1 + a_2x_2 + a_3x_1^2$. The coordinate vector z stores the values of those functions along a trajectory. A dictionary is closed under differentiation along the dynamics when each observable's derivative is in that same span.

The closure graph in Figure 1.2 is a quick visual test. An arrow into a node means that the indicated observable contributes to that node's derivative. Every arrow originates at a retained node. No fourth function is needed. Many useful dictionaries fail this test; their missing functions create the central approximation problem of the book.

Free Sample



Arrows show terms in each observable's derivative.

Figure 1.2: Derivative closure for the three-observable dictionary. The source x_1^2 drives x_2, while each observable has its own decay term. All derivative dependencies stay inside the retained set. Arrows describe coordinate equations, not extra physical interactions.

1.3 Initialization, evolution, and readout are separate steps

A matrix equation alone is not yet a model of the original system. We must specify how a physical state becomes a lifted state and how to recover the desired output. Here the state readout is linear:

$$x = Cz, \qquad C = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{pmatrix}. \tag{1.6}$$

For a chosen initial state $x(0)$, the complete prediction is

$$\widehat{x}(t) = Ce^{tA}\psi(x(0)). \tag{1.7}$$

The hat denotes a model output. In this particular example it equals the true state exactly, in exact arithmetic.

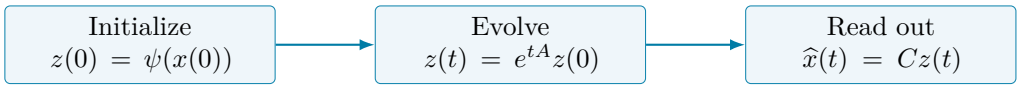
What does a matrix exponential mean?

For a scalar equation $\dot{v} = av$, the solution is $e^{at}v(0)$. For a constant square matrix A, define

$$e^{tA} = I + tA + \frac{t^2A^2}{2!} + \cdots.$$

The series converges for every finite matrix and finite t. Termwise differentiation gives $\frac{\mathrm{d}}{\mathrm{d}t}e^{tA} = Ae^{tA}$, and its value at zero is I. Thus $z(t) = e^{tA}z(0)$ solves the matrix initial-value problem. It is generally false that exponentiating each entry of A produces e^{tA}. The lab uses a matrix-exponential routine.

Free Sample



The nonlinear initialization must be consistent with the physical state.

Figure 1.3: A lifted predictor has three specified maps: initialize from the physical state, evolve the coordinates, and read out the requested output. Only the middle map must be linear in the lifted state. Here the readout is also linear, and the initialization contains the square.

Proposition 1.1 (An exact lifted predictor)**.** *Let $\kappa_1, \kappa_2 < 0$, let ψ be given by Equation (1.4), and let A, C be given by Equations (1.5) and (1.6). For every $x(0) \in \mathbb{R}^2$ and every $t \geq 0$, the solution of Equation (1.1) is $Ce^{tA}\psi(x(0))$.*

Proof. The first component is $x_1(t) = e^{\kappa_1 t}x_1(0)$. The third lifted component, initialized at $x_1(0)^2$, is $z_3(t) = e^{2\kappa_1 t}x_1(0)^2 = x_1(t)^2$. The second lifted equation is therefore the same forced scalar equation as the physical x_2 equation, with the same initial value. Uniqueness for that linear forced equation gives $z_2(t) = x_2(t)$. Both components are defined for every $t \geq 0$ by these exponential formulas. □

This proof does not assume distinct rates. If $\kappa_2 = 2\kappa_1$, a formula obtained by dividing by $2\kappa_1 - \kappa_2$ must be replaced by its limit, but the matrix equation remains valid. Resonance can make an eigenvector representation inconvenient without breaking the lifted model. Later chapters develop that distinction.

1.3.1 The lifted surface

Not every vector in $\mathbb{R}^3$ is a consistent lifted state. Physical states give the surface $z_3 = z_1^2$, shown in Figure 1.4. To check that evolution preserves this relation, define the consistency defect $d(t) = z_3(t) - z_1(t)^2$. Then

$$\dot{d} = 2\kappa_1 z_3 - 2z_1\kappa_1 z_1 = 2\kappa_1 d.$$

Consequently $d(0) = 0$ implies $d(t) = 0$ for all subsequent times.

If we instead initialize $z_1 = 1, z_2 = 0, z_3 = 0$, the linear model predicts zero second coordinate. The lifted matrix still evolves a perfectly valid vector, but that vector did not correspond to the chosen physical state. The error came from initialization, not from the matrix dynamics. This issue will reappear when control algorithms

Free Sample

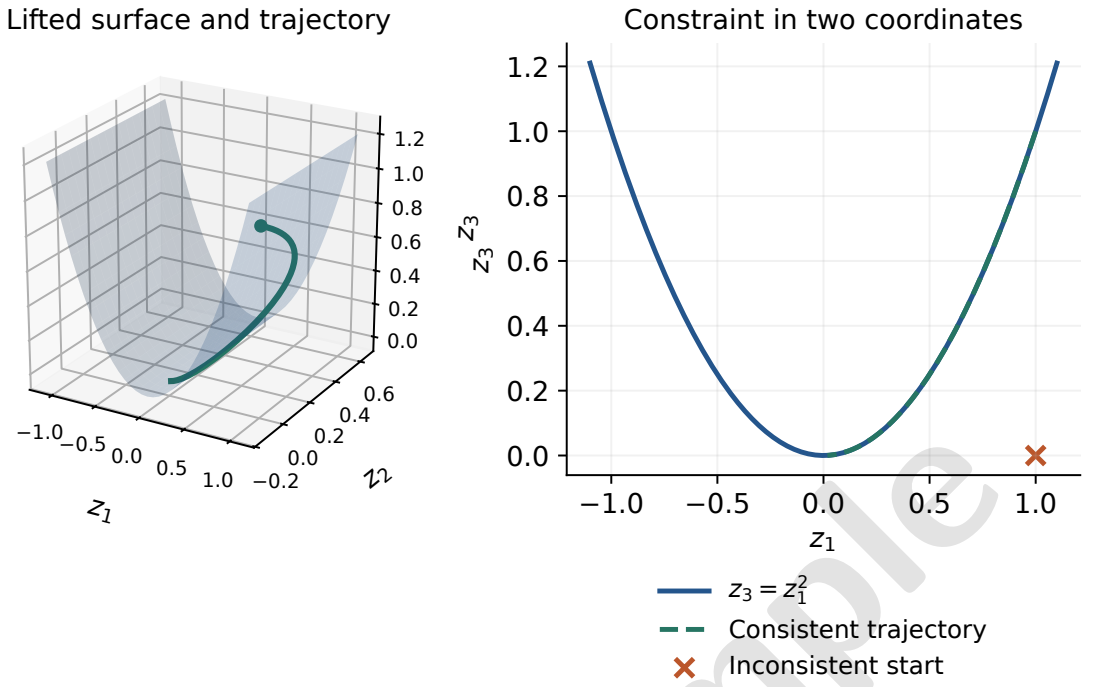


Figure 1.4: Consistent lifted states lie on the curved surface $z_3 = z_1^2$. A lifted trajectory initialized on it stays on it. The ambient linear system also accepts points off the surface; these do not have the specified physical interpretation. The two-dimensional panel exposes the constraint without the perspective of the three-dimensional view.

repeatedly reset lifted states and when compression drops coordinates that carry initial-state effects.

The lift does not imply superposition in the original state. Usually $\psi(x + \tilde{x}) \neq \psi(x) + \psi(\tilde{x})$ because squaring does not distribute over addition. The matrix flow is linear on z; the physical initial-state-to-trajectory map remains nonlinear.

1.4 Why functions can evolve linearly

The example might look like a fortunate trick. There is a broader reason to try functions as coordinates. Let $\varphi^t(x)$ mean the state reached after time t from initial state x, whenever that solution exists. If g is a scalar measurement, its value at that later state is $g(\varphi^t(x))$. Define the evolution of the measurement by

$$(\mathcal{K}^t g)(x) = g(\varphi^t(x)). \tag{1.8}$$

Free Sample

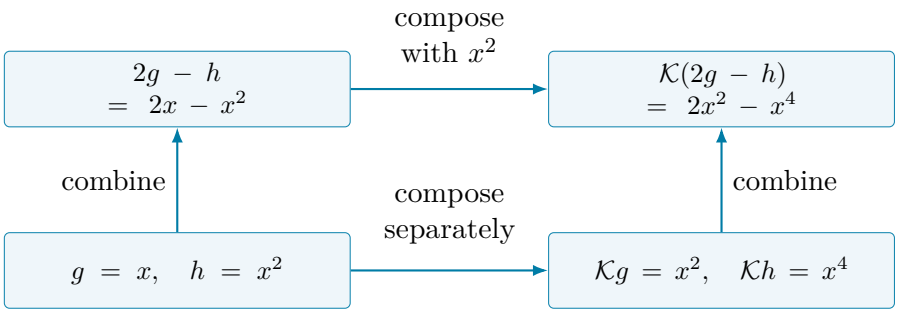


Figure 1.5: Linearity acts on observables. Evolving the combination $2g - h$ gives the same function as evolving g and h separately and combining their results. The state update $x \mapsto x^2$ is nonlinear in x. The extra x^4 term also shows why a chosen finite dictionary can fail to be closed.

The map $\mathcal{K}^t$ takes a function as input and produces a function. For constants a, b and functions g, h,

$$\begin{aligned}[\mathcal{K}^t(ag + bh)](x) &= a\, g(\varphi^t(x)) + b\, h(\varphi^t(x)) \\ &= a[\mathcal{K}^t g](x) + b[\mathcal{K}^t h](x).\end{aligned}$$

This is linearity in functions. It holds even when φ^t is nonlinear in the initial state. It does not require or assert $\varphi^t(ax + b\widetilde{x}) = a\varphi^t(x) + b\varphi^t(\widetilde{x})$. This linear function-evolution map is called a *Koopman operator*. For now, the displayed composition and its addition rule are all we need.

Example 1.2 (A one-step calculation)**.** For the map $x^+ = x^2$ on $[-1, 1]$, let $g(x) = x$ and $h(x) = x^2$. The one-step operator satisfies $\mathcal{K}g = x^2$ and $\mathcal{K}h = x^4$. For the combined measurement $2g - h$, direct substitution gives $\mathcal{K}(2g - h) = 2x^2 - x^4 = 2\mathcal{K}g - \mathcal{K}h$. The operator is linear, but the span of x and x^2 is not invariant: the function x^4 is missing. Adding x^4 creates x^8 at the next step. Linearity alone has not given a finite closed dictionary.

For differentiable g and an ODE $\dot{x} = f(x)$, the chain rule gives the instantaneous derivative

$$(\mathcal{L}g)(x) = \nabla g(x) \cdot f(x). \tag{1.9}$$

The gradient ∇g is the vector of partial derivatives of g. The map $\mathcal{L}$ is called the *generator*. It is linear in g, because differentiation and the dot product with a fixed vector field preserve constant linear combinations. It need not map a chosen finite

Free Sample

span into itself. Our three coordinates work because each $\mathcal{L}\psi_j$ is in their span. This is what the earlier derivative calculation checked. General claims about generators also require a specified function space and domain; the elementary calculation here uses only differentiable functions evaluated along the explicit ODE solution.

1.5 What happens when closure fails?

Consider a different scalar system,

$$\dot{x} = -x + x^2.$$

Trying the coordinates (x, x^2) gives

$$\frac{\mathrm{d}}{\mathrm{d}t}x = -x + x^2, \qquad \frac{\mathrm{d}}{\mathrm{d}t}x^2 = -2x^2 + 2x^3.$$

The new term x^3 is not a constant linear combination of x and x^2 on an interval. Adding it produces x^4, and so on. A hierarchy of infinitely many coordinates can be exact even when each finite cut is approximate. Chapter 6 derives its truncation error. We make no all-time or all-state approximation claim here.

There are several responses to a missing term. We can set it to zero, approximate it from the retained coordinates, or choose a different dictionary so that less important information is missing. If unresolved variables feed back after some delay, their influence may be better represented through memory. Chapter 9 derives that memory without invoking an infinite-dimensional identity.

A useful finite model also depends on the requested output. In Equation (1.1), the single observable x_1 predicts x_1 exactly. It cannot recover x_2 for arbitrary initial states, since two states with the same x_1 can have different x_2. The span of x_1 and x_1^2 is closed as well, but still omits that information. The three-coordinate dictionary predicts both physical states. Counting coordinates without naming the output cannot settle which model is sufficient.

Free Sample

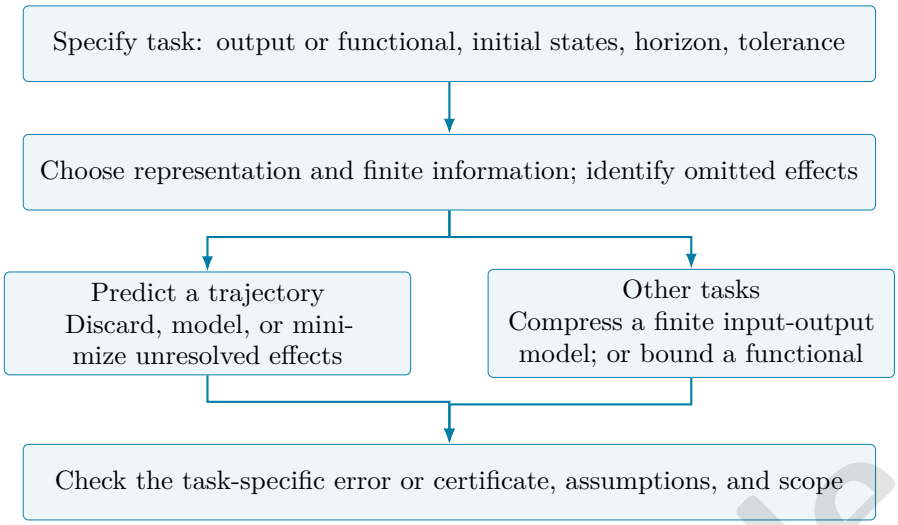


Figure 1.6: Task-first modeling. The central prediction branch addresses unresolved effects by discarding, modeling, or minimizing them. Compression and bounding answer related but distinct questions. All branches return to a statement of scope and an appropriate check. This is a schematic decision aid, not a claim that the listed operations are interchangeable.

1.6 A workflow for the rest of the book

Figure 1.6 organizes the choices. Start by writing the task: for example, predict x_2 up to time four for a specified set of initial states. Choose how to represent the dynamics, which finite information to keep, and how to obtain the coefficients. Then derive or measure what the retained model omits. Finally test the requested output on the intended states and horizon.

For a prediction task, three broad actions are available. *Discard* omitted effects when their propagated influence is acceptable. *Model* them through closure, memory, or auxiliary variables. *Minimize* them by improving coordinates or the projection, while recognizing that a small residual in a training norm need not guarantee a small future error. These actions can be combined.

Two complementary branches concern different tasks. *Compress* an already finite input-output model when fewer internal states preserve the responses of interest. *Bound* an accumulated quantity using balance laws and positivity when a full trajectory is unnecessary. Neither branch is automatically a better predictor, and each needs its own assumptions.

Free Sample

This book develops these choices using observables, polynomial coordinates, and distributions. Stochastic and distributed systems enter after their probability and PDE background. Linear control ideas enter after we introduce inputs. The unifying question remains concrete: what information does the finite model retain, and how does the information it omits affect the task?

Checks before trusting a lift

State the dictionary, initialization rule, readout, initial-state set, and time horizon. Differentiate every retained observable. Identify any term outside the retained span. Check an independent trajectory or analytic formula. Distinguish algebraic exactness from numerical accuracy, and avoid extending a favorable example into a generic performance claim.

1.7 Guided lab: three routes to the same trajectory

The supplied script is a reference check at its declared defaults. For a changed experiment, use a working copy and distinguish algebraic, solver, and domain checks from expectations specific to those defaults. The numerical-practice appendix explains how to preserve both kinds. The executable companion is `teaching/labs/lab01_exact_lift.py`. It supplies NumPy arrays for vectors and matrices, SciPy's ODE integrator for the nonlinear reference, and a matrix exponential for the lifted model. Run it from the project root. No fitted parameters or training data are used. It compares three independently specified computations:

1. Integrate Equation (1.1) with a general ODE solver.
2. Evaluate $Ce^{tA}\psi(x(0))$ on the same time grid.
3. Evaluate the closed form in Equation (1.3).

The output reports maximum absolute errors over the sampled times. The nonlinear reference uses the DOP853 method with relative tolerance 10^{-11} and absolute tolerance 10^{-13}. The checks require agreement within 10^{-9}, leaving room above the numerical discrepancy. Solver tolerances control local numerical error estimates; they do not prove an algebraic identity or supply a certified uniform error bound.

Free Sample

First read the script and predict which component the tangent model will miss. Next run it and inspect the reported errors. Finally change only the initial x_1 to ε and keep $x_2(0) = 0$. The exact second state becomes $3\varepsilon^2(e^{-2t} - e^{-3t})$. Check the maximum error for $\varepsilon = 1, 1/2, 1/4$. Its quadratic scaling explains why the tangent model improves locally while omitting the same physical mechanism. The script prints these three errors.

The lab is a transparent analytic-reference check. It does not test identification under noise, large dictionaries, computational savings, or performance on an unseen class of nonlinear systems.

1.8 Exercises

1.1. Practice. Derive all three rows of Equation (1.5) without substituting numerical rates. Which use of the chain rule makes the third row close?

1.2. Practice. Differentiate Equation (1.3) and verify the peak time and value. What is the maximum absolute error of the tangent prediction for x_2 over $[0, 4]$? How does it scale with $x_1(0) = \varepsilon$?

1.3. Practice. Set $z(0) = (1, 0, 0)^\top$ in the lifted matrix equation. Solve its second and third components. Explain why the resulting zero output is consistent with the matrix equation but inconsistent with the prescribed physical initialization.

1.4. Connect. Derive the consistency-defect equation $\dot{d} = 2\kappa_1 d$. For arbitrary $d(0)$, find $d(t)$. Does a decaying defect mean the initialization was physically exact?

1.5. Connect. For Equation (1.1), decide whether each dictionary is closed: (x_1), (x_1, x_2), (x_1, x_1^2), $(x_1, x_2, x_1^2, x_1x_2)$. For each closed dictionary, state whether it can linearly read out both physical states for arbitrary initial values.

1.6. Connect. For $x^+ = x^2$, compute $\mathcal{K}(3x + 2x^2)$ and $\mathcal{K}(x^4)$. Verify the linear addition rule and explain why it does not assert that the state update is linear.

Free Sample

1.7. Connect. For $\dot{x} = -x + x^2$, show that $\frac{\mathrm{d}}{\mathrm{d}t}x^j = -jx^j + jx^{j+1}$ for every positive integer j. Write the three-coordinate model obtained by setting x^4 to zero. Identify exactly which equality has become an approximation.

1.8. Explore. Solve the second state of Equation (1.1) when $\kappa_2 = 2\kappa_1$, with general initial values. Explain why a factor t appears even though the lifted matrix is constant.

1.9. Explore. A desired output is x_1^4, rather than the physical state. Construct a one-coordinate exact linear model for it. What is its initialization? Why does it not recover the sign of x_1?

1.10. Explore. Write a model specification for predicting the maximum of x_2 on a finite interval. Name the initialization set, output, horizon, error metric, and an independent validation check. Identify one claim that a single successful trajectory would not establish.

1.9 Notes and further reading

Koopman's original construction establishes linear evolution on observables (Koopman, 1931). Finite invariant observable subspaces and their relation to state prediction are discussed by Brunton et al. (2016); broader modern context is supplied by Brunton et al. (2022). Differentiating polynomial coordinates leads to the classical Carleman hierarchy (Carleman, 1932). These sources provide context for the constructions, not a prerequisite for the calculations in this chapter. The distinctions among initialization, evolution, and readout are essential when interpreting any such model. Our worked derivations are expository instances of classical ideas, not new linearization theorems.

Free Sample

Chapter 2

Linear dynamics and the matrix tools we need

A linear model is useful only if we know how to interpret its matrix. Negative eigenvalues, for example, do not guarantee that every disturbance shrinks immediately. Repeated eigenvalues need not provide independent eigenvectors. An accurate coordinate evolution can also become inaccurate after an ill-conditioned change of coordinates. These are practical issues for lifted models, not exceptions to linear algebra.

What you will be able to do

Solve a small forced matrix equation; translate an equilibrium to zero; interpret decay and oscillation modes; handle a Jordan block; distinguish long-time stability from transient amplification; and use norms, singular values, and condition numbers to describe sensitivity.

2.1 A two-body thermal calculation

Let θ_1, θ_2 be temperature deviations from an ambient value. Suppose each body loses heat to the environment and exchanges heat with the other. After scaling heat capacities and time, a simple model is

$$\dot{\theta}_1 = -2\theta_1 + \theta_2 + u, \qquad \dot{\theta}_2 = \theta_1 - 2\theta_2. \tag{2.1}$$

Free Sample

The input u is heating applied to the first body. The model assumes linear heat-transfer laws and constant coefficients. It is a teaching model, not a statement that all thermal systems are linear.

Set $u = 1$. An equilibrium has zero derivatives, so $2\theta_1 - \theta_2 = 1$ and $\theta_1 - 2\theta_2 = 0$. Solving gives $\theta_* = (2/3, 1/3)^\top$. Introduce the deviation $v = \theta - \theta_*$. Because θ_* solves the steady equation,

$$\dot{v} = Av, \qquad A = \begin{pmatrix} -2 & 1 \\ 1 & -2 \end{pmatrix}.$$

This translation turns a constant-forced equation into a homogeneous one. It does not change the physical equilibrium, only the coordinate origin.

The vectors $q_+ = (1, 1)^\top$ and $q_- = (1, -1)^\top$ satisfy $Aq_+ = -q_+$ and $Aq_- = -3q_-$. They are *eigenvectors*; their multiplying constants are *eigenvalues*. Equal temperature deviations decay with rate -1; the difference between the bodies decays with rate -3 because heat exchange also removes their imbalance. Every vector in $\mathbb{R}^2$ can be written uniquely as $a_+q_+ + a_-q_-$, giving

$$v(t) = a_+e^{-t}q_+ + a_-e^{-3t}q_-.$$

For initially ambient bodies, $\theta(0) = (0, 0)^\top$, $a_+ = -1/2$ and $a_- = -1/6$. Therefore

$$\begin{aligned} \theta_1(t) &= \tfrac{2}{3} - \tfrac{1}{2}e^{-t} - \tfrac{1}{6}e^{-3t}, \\ \theta_2(t) &= \tfrac{1}{3} - \tfrac{1}{2}e^{-t} + \tfrac{1}{6}e^{-3t}. \end{aligned} \tag{2.2}$$

Figure 2.1 shows both temperatures and the two decaying contributions. The input creates a nonzero equilibrium; the eigenvalues describe deviations about that equilibrium.

2.2 Matrix exponentials and forcing

For any real square matrix A, the matrix exponential introduced in Chapter 1 solves $\dot{v} = Av$: $v(t) = e^{tA}v(0)$. The identity matrix I leaves vectors unchanged, so $e^{0A} = I$. The same matrix is used at each time, and the series definition gives $e^{(t+s)A} = e^{tA}e^{sA}$. This composition rule expresses evolving first for s and then for t.

Free Sample

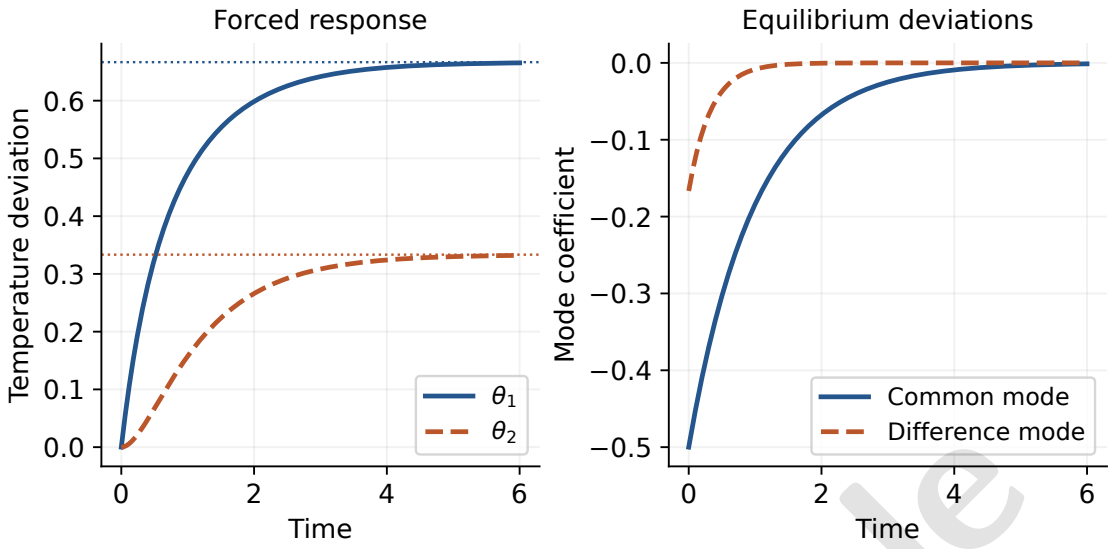


Figure 2.1: A forced thermal response separates into an equilibrium and two decay modes. The initial condition is zero and the heating input is one. The slow common mode and faster difference mode explain the different approach of the two temperatures to their steady values.

Proposition 2.1 (Variation of constants)**.** *Let A be a constant $n \times n$ matrix and let $r(t)$ be a continuous vector on $[0, T]$. The unique solution of $\dot{v} = Av + r(t)$ with initial value v_0 is*

$$v(t) = e^{tA} v_0 + \int_0^t e^{(t-s)A} r(s)\, \mathrm{d}s. \tag{2.3}$$

Proof. Multiply the equation by e^{-tA} and differentiate the product. The terms containing Av cancel, leaving $\frac{\mathrm{d}}{\mathrm{d}t}(e^{-tA}v) = e^{-tA} r(t)$. Integration gives the formula. Differentiating it verifies both the equation and initial condition. The difference of two solutions, multiplied by e^{-tA}, has zero derivative and zero initial value, proving uniqueness. □

There is another useful route to the exponential. A forward Euler step with step size h is $v_{j+1} = (I + hA)v_j$. After m steps covering time t, its prediction is $(I + tA/m)^m v_0$, which tends to $e^{tA} v_0$ as m grows. For a finite step, however, this is an approximation. Even scalar decay $\dot{v} = -v$ becomes unstable under Euler if $|1 - h| > 1$. Stability of the differential equation and stability of a numerical time-stepping method are separate questions.

Free Sample

2.3 Modes and changes of coordinates

If A has n independent eigenvectors, collect them as columns of a matrix V. Then $AV = V\Lambda$, where Λ is diagonal. Writing $v = Va$ gives $\dot{a} = \Lambda a$: each component evolves by a scalar exponential. Transforming back gives

$$e^{tA} = Ve^{t\Lambda}V^{-1}.$$

The inverse exists because the columns are independent. This formula explains diagonalization; it is not the definition of the exponential. The exponential exists even when no such V exists.

For the thermal matrix, normalize the eigenvectors to length one by dividing them by $\sqrt{2}$. The resulting V obeys $V^\top V = I$. Such a matrix is *orthogonal*: it preserves Euclidean lengths and angles. In this example the mode coordinates do not amplify measurement errors. A general eigenvector matrix need not have this property.

2.3.1 Oscillations and complex eigenvalues

Consider

$$A = \begin{pmatrix} -\gamma & -\omega \\ \omega & -\gamma \end{pmatrix}, \qquad \gamma \geq 0.$$

Its real trajectory is

$$v(t) = e^{-\gamma t} \begin{pmatrix} \cos \omega t & -\sin \omega t \\ \sin \omega t & \cos \omega t \end{pmatrix} v(0).$$

The rotation matrix preserves length, while $e^{-\gamma t}$ controls the amplitude. The eigenvalues are $-\gamma \pm i\omega$, where $i^2 = -1$. Complex eigenvalues are compact bookkeeping for this pair of real sinusoidal components. Real parts describe decay or growth; imaginary parts describe oscillation frequency. A zero real part does not imply decay. Figure 2.2 displays the real motion.

Free Sample

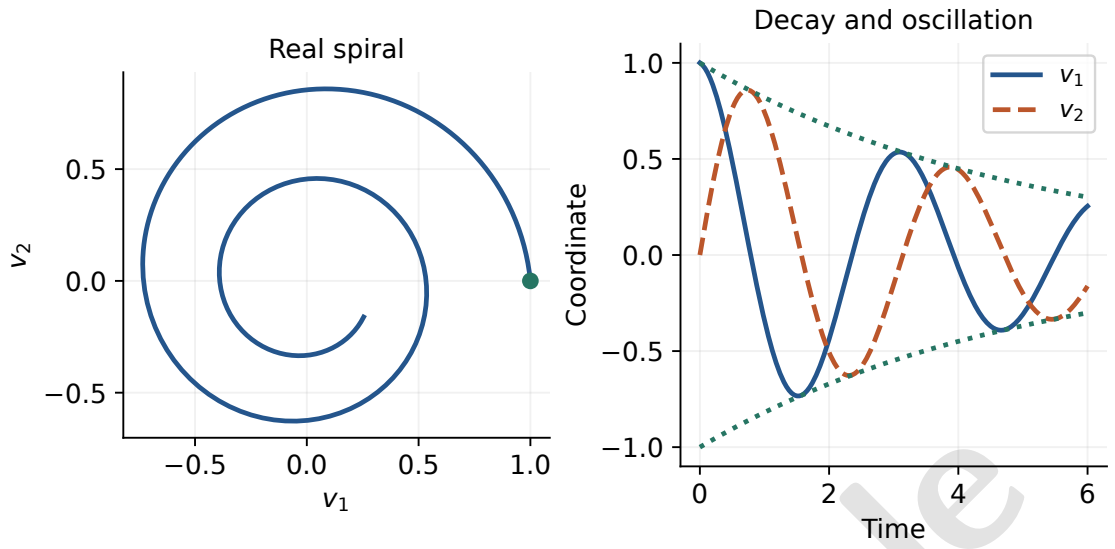


Figure 2.2: A damped rotation with $\gamma = 0.2$, $\omega = 2$, and $v(0) = (1, 0)^\top$. A complex eigenvalue pair describes a real spiral. The coordinate oscillations remain inside the amplitude envelopes $\pm e^{-0.2t}$.

2.4 Repeated rates and Jordan blocks

Take

$$J = \begin{pmatrix} -1 & 1 \\ 0 & -1 \end{pmatrix} = -I + N, \qquad N = \begin{pmatrix} 0 & 1 \\ 0 & 0 \end{pmatrix}.$$

Both eigenvalues are -1, but an eigenvector must have second component zero. There is only one independent eigenvector. Since $N^2 = 0$, the exponential series for N stops after its linear term. Because I commutes with N,

$$e^{tJ} = e^{-t}(I + tN) = e^{-t} \begin{pmatrix} 1 & t \\ 0 & 1 \end{pmatrix}.$$

Starting from $(0, 1)^\top$ gives $v_1(t) = te^{-t}$ and $v_2(t) = e^{-t}$. The first component rises before decaying, despite identical negative eigenvalues. A matrix of this form is a *Jordan block*. Figure 2.3 compares the response with an uncoupled repeated rate. This mechanism explains the factor t at resonance in the lifted system of Chapter 1.

Free Sample

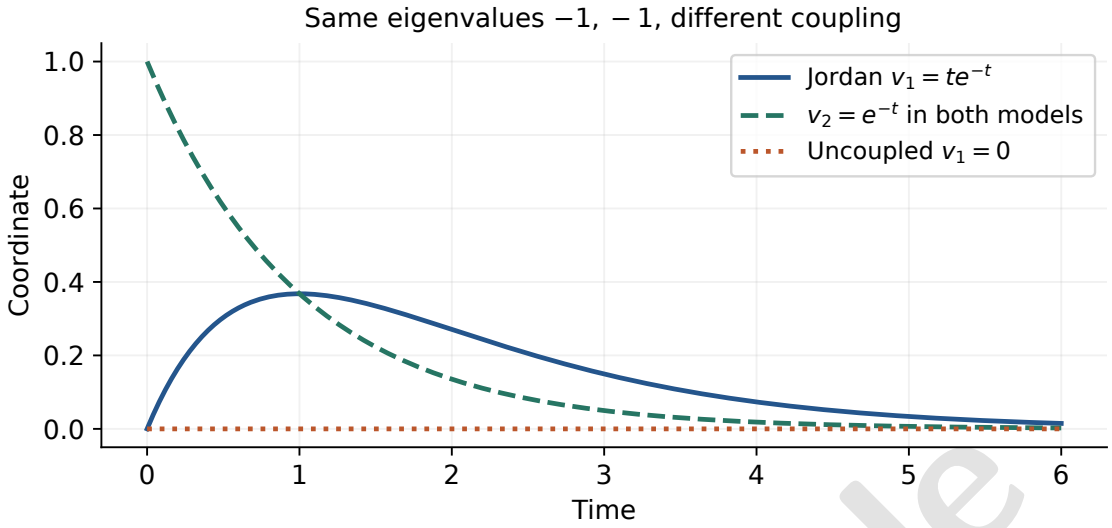


Figure 2.3: Repeated eigenvalues do not remove coupling. The Jordan system initialized at $(0, 1)^\top$ has a response te^{-t} in its first coordinate. The uncoupled matrix $-I$ has the same eigenvalues but keeps that coordinate zero. The distinction is visible in the time response.

2.5 Norms, singular values, and conditioning

The Euclidean length of v is $\|v\|_2 = (\sum_j v_j^2)^{1/2}$. For a matrix M, its induced norm is the largest factor by which it can stretch a vector:

$$\|M\|_2 = \max_{\|v\|_2=1} \|Mv\|_2.$$

The triangle inequality and this definition imply $\|Mv\|_2 \le \|M\|_2\|v\|_2$ and $\|MN\|_2 \le \|M\|_2\|N\|_2$. These simple inequalities turn omitted forcing into error estimates later.

To calculate the stretch, square its length: $\|Mv\|_2^2 = v^\top M^\top M v$. The matrix $M^\top M$ is symmetric and has nonnegative eigenvalues, since its quadratic form is a squared length. A real symmetric matrix has an orthonormal eigenbasis. In that basis the quadratic form is a weighted sum of squared coordinates, so its largest value on the unit sphere is its largest eigenvalue. The square roots of these eigenvalues are the *singular values* of M. The largest singular value is therefore $\|M\|_2$. The corresponding input directions are its right singular vectors. They describe stretching, which is a different question from the eigenvector question $Mv = \lambda v$ about directions mapped onto themselves.

Free Sample

Example 2.2 (An ellipse and a sensitive inverse)**.** The matrix $M = \operatorname{diag}(3, 1/10)$ maps the unit circle to an ellipse with semiaxes 3 and $1/10$. Its largest stretch is 3 and its smallest is $1/10$. Inverting the compressed direction amplifies an absolute error by ten. For an invertible matrix define its *condition number* by $\operatorname{cond}_2(M) = \|M\|_2 \|M^{-1}\|_2$. Here it is 30. The ratio of the largest to smallest singular value gives the same number. A singular matrix has no inverse and an infinite condition number by convention.

If $Mv = b$ and the measured right side is $b + \delta b$, then $\delta v = M^{-1}\delta b$. When b, v are nonzero,

$$\frac{\|\delta v\|_2}{\|v\|_2} \le \operatorname{cond}_2(M) \frac{\|\delta b\|_2}{\|b\|_2}.$$

To see this, first bound $\|\delta v\|_2$ by $\|M^{-1}\|_2 \|\delta b\|_2$, then use $\|b\|_2 \le \|M\|_2 \|v\|_2$. The estimate concerns a perturbation in b with M fixed. Perturbing the matrix itself requires an additional argument.

2.6 Stable eigenvalues can coexist with transient growth

Consider a different matrix,

$$A = \begin{pmatrix} -1 & 10 \\ 0 & -2 \end{pmatrix}, \qquad e^{tA} = \begin{pmatrix} e^{-t} & 10(e^{-t} - e^{-2t}) \\ 0 & e^{-2t} \end{pmatrix}. \tag{2.4}$$

Both eigenvalues are negative. Starting at $(0, 1)^\top$, however, the first component reaches $5/2$ at $t = \log 2$, while the second is $1/4$. The state norm has grown above $5/2$ from its initial value one. The full solution eventually decays. *Asymptotic stability* of a constant matrix means all trajectories approach zero; it does not require monotone decay of their norm.

This matrix is not normal: $A^\top A \neq A A^\top$. Nonorthogonal modes can cancel initially and later cease to cancel. Figure 2.4 shows an individual response and the largest possible stretch $\|e^{tA}\|_2$. An exponential error estimate often needs a prefactor $M > 1$ even if its long-time rate is negative.

Free Sample

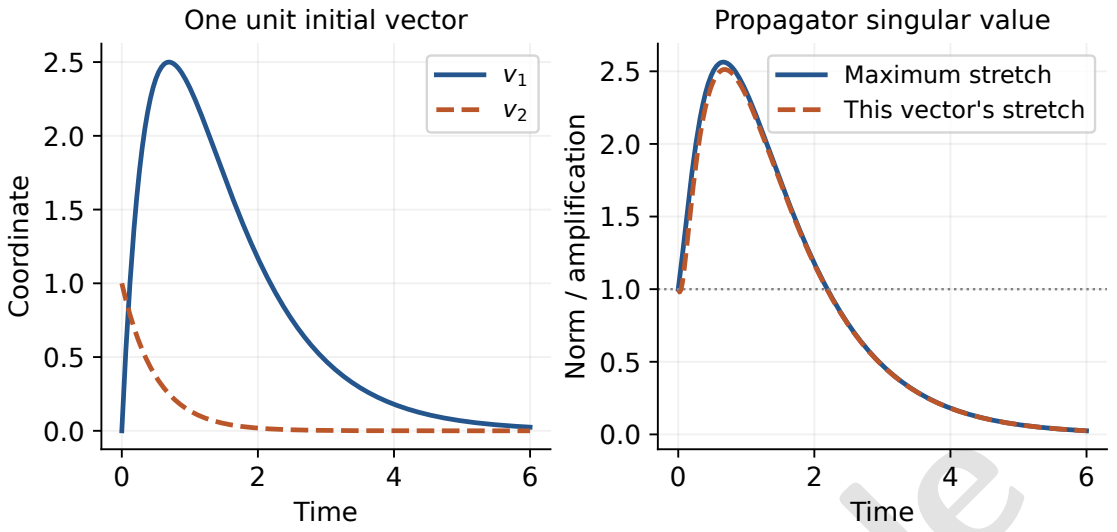


Figure 2.4: Transient amplification for Equation (2.4). The response from $(0, 1)^\top$ grows before decaying. The induced norm of the propagator gives the maximum amplification over all unit initial vectors at each time. The negative eigenvalues alone do not give this curve.

For contrast, if the symmetric part $(A + A^\top)/2$ is negative semidefinite, then

$$\frac{\mathrm{d}}{\mathrm{d}t}\|v(t)\|_2^2 = v(t)^\top (A + A^\top) v(t) \leq 0.$$

This stronger condition guarantees nonincreasing Euclidean norm. It depends on the chosen norm and coordinates. We will call this a dissipativity calculation, and use it with its stated inner product.

2.7 Guided lab: solve, change coordinates, and measure amplification

The supplied script is a reference check at its declared defaults. For a changed experiment, use a working copy and distinguish algebraic, solver, and domain checks from expectations specific to those defaults. The numerical-practice appendix explains how to preserve both kinds. Run `teaching/labs/lab02_linear.py`. It compares the thermal formula with an independently integrated forced ODE, checks the explicit Jordan and triangular exponentials against a matrix-exponential routine, and calculates propagator norms using singular values. The ODE reference uses

Free Sample

DOP853 with relative tolerance 10^{-11} and absolute tolerance 10^{-13}; the maximum sampled discrepancy must be below 10^{-9}. Algebraic identities checked by matrix arithmetic use a 10^{-12} threshold.

Before running, predict the initial derivatives of the two temperatures. Then verify them from the formula. Change the triangular coupling from ten to zero and explain why the growth disappears. Finally sample the unit circle and compare its largest observed stretch with the largest singular value at a fixed time. Finite angular sampling approaches the maximizing direction but does not replace the definition of the norm. The supplied figures are exact-formula demonstrations, not performance tests of a fitted method.

From the bundle root, use `.venv/bin/python teaching/labs/lab02_linear.py -coupling 0` for the uncoupled variation.

2.8 Exercises

2.1. Practice. Derive the thermal equilibrium and both mode coefficients for zero initial temperature deviations and $u = 1$. Verify the initial derivatives directly.

2.2. Practice. For constant heating $u = u_0$, derive the equilibrium and response from zero. Which part of the solution scales with u_0?

2.3. Practice. Verify Equation (2.3) by differentiating its right side. Why does the upper endpoint of the integral contribute $r(t)$?

2.4. Connect. Compute the eigenvectors of the thermal matrix and normalize them. Show that their coordinate transformation preserves Euclidean length.

2.5. Connect. Derive the damped-rotation formula by differentiating it. For $\gamma = 0$, what is conserved and what fails to approach zero?

2.6. Connect. Derive e^{tJ} from $N^2 = 0$ and find the time and value of the maximum of te^{-t}.

2.7. Connect. For $\dot{v} = -v$, find the range of positive Euler step sizes that give decay of the sampled solution. Discuss the boundary values.

Free Sample

2.8. Explore. For $M = \mathrm{diag}(3, 1/10)$, calculate its singular values and condition number. Give a right-side perturbation whose absolute amplification under M^{-1} is ten.

2.9. Explore. Derive the off-diagonal entry in Equation (2.4) by solving the second equation first. Verify the state and its norm at $t = \log 2$.

2.10. Explore. Let $A = \mathrm{diag}(-1, -2)$. Prove directly that $\|e^{tA}\|_2 = e^{-t}$ for $t \geq 0$. Explain why the analogous claim is false for Equation (2.4).

2.9 Notes and further reading

The matrix tools developed here underpin linear model reduction; a systematic later treatment is given by Antoulas (2005). For now the important habits are to compute a response, name a norm, and separate eigenvalue conclusions from amplification conclusions. No result in this chapter requires a lifted model to be learned from data.

Free Sample

Chapter 3

Functions as coordinates

A laboratory may provide a temperature, a squared voltage, a threshold event, or a camera-derived angle. Each is a function of the underlying state. Before asking how these measurements evolve, we should ask what they distinguish. Two different states that give the same coordinate vector cannot be distinguished by a readout of that vector alone.

What you will be able to do

Choose and evaluate a dictionary; recognize dependent features; explain why affine models need a constant coordinate; test whether a desired output can be recovered; and distinguish a linear readout from a nonlinear decoder and from a constrained lifted initialization.

3.1 A dictionary is a bank of functions

Let $x \in X \subseteq \mathbb{R}^n$ denote a physical state. A dictionary is an ordered list of real-valued functions on X,

$$\psi(x) = (\psi_1(x), \dots, \psi_N(x))^\top.$$

Its value $z = \psi(x)$ is the coordinate vector supplied to a model. The functions may come from sensors or be evaluated from a measured state. The order matters when assembling matrices; the mathematical span does not depend on that order. Figure 3.1 separates the function definitions from the values evaluated at one state.

Free Sample

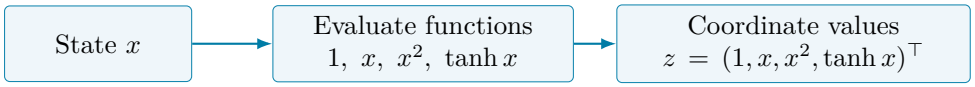


Figure 3.1: A state is evaluated by several functions to form one coordinate vector. The entries may have different units and scales. Calling them coordinates does not assert that they uniquely determine the state or that their dynamics close in a finite span.

For a scalar state, examples include $1, x, x^2$, a saturating sensor $\tanh x$, and an indicator of a specified interval. The constant function takes value one at every state. Here $\tanh x = (e^x - e^{-x})/(e^x + e^{-x})$ is a smooth function bounded between minus one and one. An indicator is one on its set and zero elsewhere. For a pendulum angle, $\sin\theta$ and $\cos\theta$ respect its periodic nature, while a raw angle coordinate must choose where to cut the circle. A discontinuous indicator is convenient for counting samples but cannot automatically be differentiated by the smooth chain-rule formula of Chapter 1.

Units matter. Adding x and x^2 without dimensional scaling can be meaningless when x has physical units. One option is to divide each state by a reference scale before constructing monomials. Another is to attach appropriate units to each coefficient. Numerical scaling matters as well: columns ranging from 10^{-6} to 10^6 can make a regression sensitive even when its algebra is correct.

3.2 Independence belongs to a domain and to the data

Functions are linearly independent on X if the identity $\sum_{j=1}^{N} a_j \psi_j(x) = 0$ for every $x \in X$ forces every coefficient a_j to be zero. This is a statement on a specified domain, not just on observed points.

The functions $1, x, x^2$ are independent on an interval with nonempty interior: a nonzero quadratic cannot vanish throughout that interval. The list $1, x, 2x$ is dependent because its last two entries have a fixed linear relation. On the restricted set $X = \{-1, 1\}$, even $1, x, x^2$ is dependent, since $x^2 = 1$ there.

For sample states $x^{(1)}, \dots, x^{(m)}$, put one evaluated dictionary in each row of the design matrix:

$$D_{ij} = \psi_j(x^{(i)}), \qquad D \in \mathbb{R}^{m \times N}. \tag{3.1}$$

Free Sample

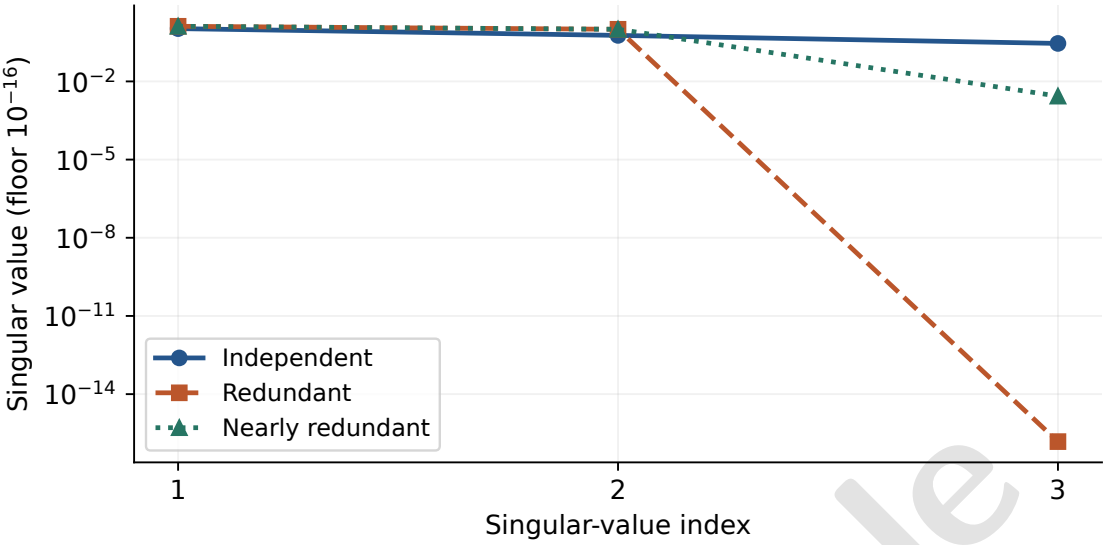


Figure 3.2: Feature dependence and sampling are distinct issues. Singular values of $D/\sqrt{m}$ are shown for uniformly spaced samples on $[-1, 1]$. The dictionary $(1, x, 2x)$ is exactly redundant; replacing the last entry by $2x + 0.02x^2$ gives a small nonzero singular value. The dictionary $(1, x, x^2)$ has independent columns on this sample set.

If a column combination is zero on the samples, the data cannot identify its coefficients uniquely. Domain independence does not prevent this. For example, three independent quadratic features evaluated only at $x = 0$ have rank one. At least N suitably placed sample states are needed for N independent columns, and the count alone is insufficient.

Example 3.1 (Exactly redundant and nearly redundant features)**.** For $\psi = (1, x, 2x)^\top$, the columns of D obey $D_{:,3} = 2D_{:,2}$. If $a = (a_1, a_2, a_3)^\top$ fits an output, then $(a_1, a_2 - 2s, a_3 + s)^\top$ fits it equally well for any s. The function predicted on the domain is unchanged, although the coefficient vector changes. With a feature $2x + \varepsilon x^2$, exact dependence disappears on an interval for $\varepsilon \neq 0$, but estimation becomes sensitive as ε approaches zero. Figure 3.2 shows the singular values that diagnose this.

Redundant functions need not be useless. Overcomplete models can be handled with regularization or a pseudoinverse, developed in Chapter 4. But an arbitrary coefficient vector from a dependent representation should not be interpreted as a uniquely identified physical law. Thresholds for numerical rank must also be stated.

Free Sample

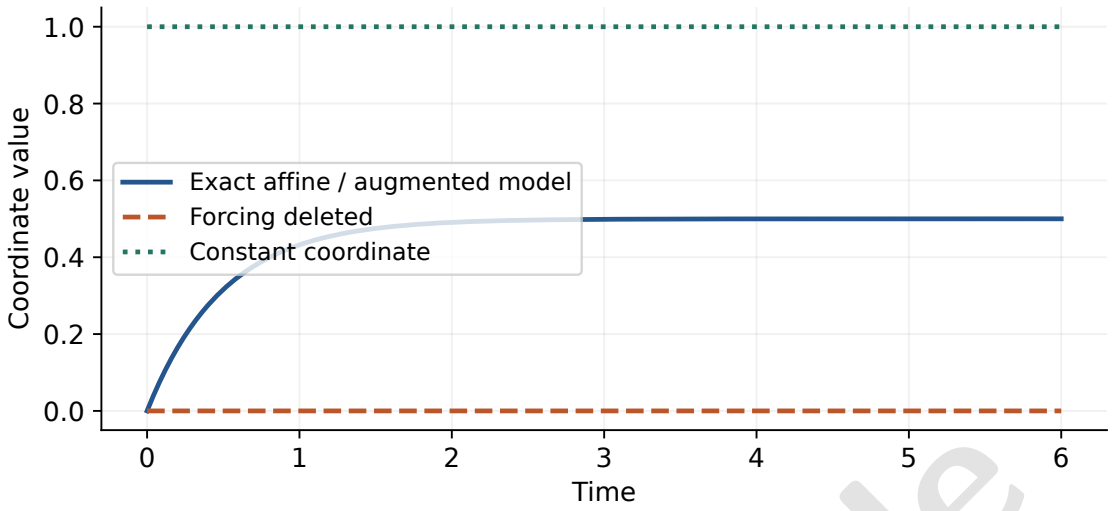


Figure 3.3: A constant coordinate preserves an affine offset. With $x_0 = 0$, the augmented model reproduces approach to $1/2$. Deleting the forcing keeps the state at zero. The additional coordinate remains identically one.

3.3 Constants turn affine equations into homogeneous linear equations

Consider a scalar sensor state obeying

$$\dot{x} = 1 - 2x.$$

It approaches $x_* = 1/2$. A homogeneous model $\dot{z} = az$ with $z = x$ cannot represent its constant derivative at $x = 0$, since $az = 0$ there. Introduce $z = (1, x)^\top$ instead. Then

$$\dot{z} = \begin{pmatrix} 0 & 0 \\ 1 & -2 \end{pmatrix} z, \qquad z(0) = (1, x_0)^\top. \tag{3.2}$$

The first coordinate has zero derivative, so it remains one and supplies the affine offset in the second equation. The solution is

$$x(t) = \tfrac{1}{2} + (x_0 - \tfrac{1}{2})e^{-2t}.$$

Figure 3.3 compares it with the equation obtained by deleting the constant forcing. Augmentation must specify the constant coordinate's initial value; setting it to zero changes the equation.

Free Sample

An alternative is the equilibrium deviation $v = x - 1/2$, which obeys $\dot{v} = -2v$. Its physical readout is $x = v + 1/2$, an affine map. Including a constant makes that readout linear on $(1, v)$. These equivalent descriptions put the offset in different places. Later, with inputs, constants and products of inputs with coordinates must be tracked just as carefully.

3.4 Can the coordinates recover the requested output?

Let $h(x)$ be the desired output. A *linear readout* is a fixed matrix C satisfying $h(x) = C\psi(x)$ on the stated domain. This is an algebraic requirement independent of how well a finite evolution matrix fits the data.

Proposition 3.2 (A necessary information test)**.** *If $\psi(x) = \psi(\widetilde{x})$ but $h(x) \neq h(\widetilde{x})$ for two states in X, no single-valued readout d can satisfy $h = d \circ \psi$ on all of X. In particular, no linear readout can do so.*

Proof. A function d applied to the same coordinate vector must return the same value. Applying it to the two hypothesized equal vectors contradicts the two different outputs. □

For $\psi(x) = x^2$ on $[-1, 1]$, the states x and $-x$ are indistinguishable. The squared state is read out perfectly, but the signed state cannot be recovered. Restricting the domain to $[0, 1]$ removes that ambiguity; the decoder $d(z) = \sqrt{z}$ then recovers the state, although not linearly. Its derivative $1/(2\sqrt{z})$ becomes large near zero. For $a, b \geq 0$ one has $|\sqrt{a} - \sqrt{b}| \leq \sqrt{|a - b|}$, but no uniform linear error-amplification bound at zero. Reconstruction conditioning is a separate issue from coordinate accuracy.

3.5 Linear evolution with a nonlinear decoder

A useful exact example is

$$\dot{x} = -x \log x, \qquad x > 0. \tag{3.3}$$

Free Sample

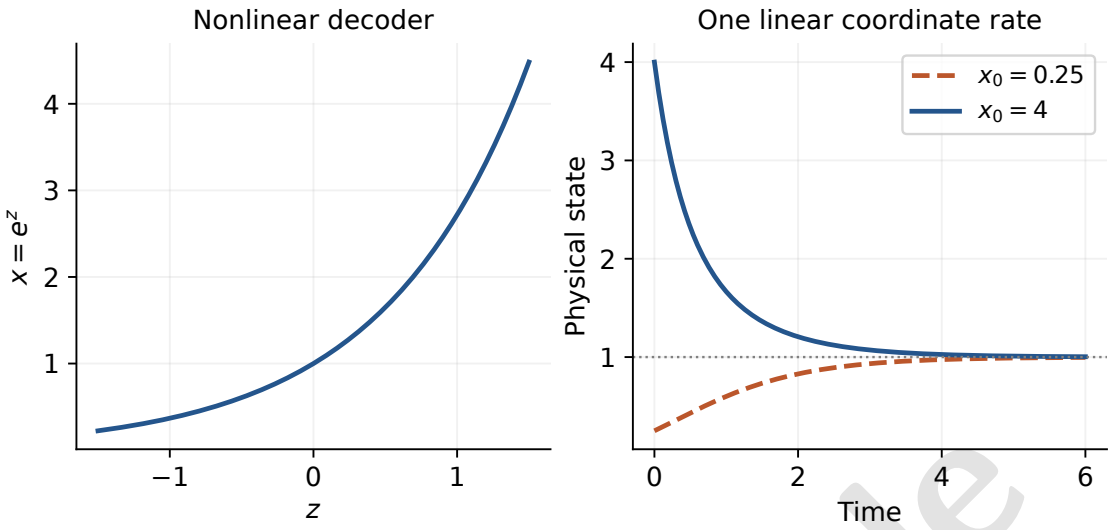


Figure 3.4: Linear coordinate evolution need not provide linear state reconstruction. For Equation (3.3), $z = \log x$ decays exponentially and the positive state is recovered by $x = e^z$. Different initial states approach one through different nonlinear state trajectories. The left panel displays the decoder itself.

Let $z = \log x$. By the chain rule, $\dot{z} = \dot{x}/x = -\log x = -z$. Thus $z(t) = e^{-t} \log x_0$ and

$$x(t) = \exp(z(t)) = x_0^{e^{-t}}.$$

This is exact linear evolution of one coordinate followed by a nonlinear decoder. It is not an exact linear state readout from z, or even from $(1, z)$, over a nontrivial interval. The exponential function is not affine in z; differentiating twice verifies that distinction.

For an approximate $\widehat{z}$, the mean value theorem gives $|e^{\widehat{z}} - e^z| \leq e^R |\widehat{z} - z|$ if both coordinates lie below R. A small coordinate error therefore has a controlled state effect only after a relevant coordinate region is bounded. Figure 3.4 shows the nonlinear readout and the state motion.

The constrained lifted surface in Chapter 1 is another geometric issue. There, (x_1, x_2, x_1^2) includes two physical states with a linear readout, but its admissible initial vectors satisfy $z_3 = z_1^2$. Here $z = \log x$ is unconstrained on $\mathbb{R}$, while the decoder is nonlinear. The examples solve different representation tasks.

Free Sample

3.6 A coordinate specification that can be audited

Before building an evolution matrix, write down the state domain, dictionary order, normalization, initialization, desired output, and readout. If data are used, also specify where sample states come from. For an indicator dictionary, record interval endpoints and what happens on boundaries. For periodic coordinates, record angle conventions. For a nonlinear decoder, identify its domain and sensitivity.

None of these choices alone guarantees closure under dynamics. Conversely, closure alone does not guarantee that a desired output is available. This separation prevents a favorable feature trajectory from being mistaken for complete state recovery.

3.7 Guided lab: feature rank, offsets, and reconstruction

The supplied script is a reference check at its declared defaults. For a changed experiment, use a working copy and distinguish algebraic, solver, and domain checks from expectations specific to those defaults. The numerical-practice appendix explains how to preserve both kinds. Run `teaching/labs/lab03_coordinates.py`. It evaluates three dictionaries on a fixed grid, computes their singular values, and checks a redundant column identity. Numerical rank uses a relative singular-value cutoff of 10^{-12}. The lab then solves the affine example by matrix exponential and checks it against its closed form. Finally it integrates Equation (3.3) from positive initial states and compares it with $\exp(e^{-t}\log x_0)$. The ODE method and tolerances are the same as in Chapter 2; discrepancies must be below 10^{-9}.

Try restricting all sample states to $\{-1, 1\}$. Which previously independent columns become dependent? Then change the logarithmic initial state from $x_0 = 4$ to $x_0 = 1/4$ and compare the state-space response with the coordinate-space response. Finally perturb z by a fixed small amount at several different values of z and measure the decoder's unequal amplification. These are controlled information and conditioning checks, not training benchmarks.

From the bundle root, use `.venv/bin/python teaching/labs/explore_labs.py -chapter 3 -initial 0.25` for these endpoint and decoder diagnostics.

Free Sample

3.8 Exercises

3.1. Practice. Give three observables for a two-state thermal system, name their units, and specify which can be evaluated from only its first sensor.

3.2. Practice. Prove dependence of $(1, x, 2x)$ on any domain. Prove independence of $(1, x, x^2)$ on an interval with nonempty interior, and explain its failure on $\{-1, 1\}$.

3.3. Practice. Construct a sample set on which independent functions x and x^2 give dependent data columns. Explain what additional samples fix it.

3.4. Connect. Verify Equation (3.2). What equation is represented if its first coordinate is initialized at two instead of one?

3.5. Connect. Write a constant-augmented representation of $\dot{x} = 3 - 4x$, and an equilibrium-deviation representation. Give both physical readouts.

3.6. Connect. Prove that x^2 cannot recover signed x on $[-1, 1]$. Give a nonlinear decoder on $[0, 1]$ and analyze it near zero.

3.7. Connect. Let $\psi = (\cos\theta, \sin\theta)^\top$ on the circle. Which angles have the same coordinate vector? Explain why this represents a circle state even though it does not select a unique real-valued angle.

3.8. Explore. Derive the logarithmic coordinate equation and its physical solution. Prove that $(1, \log x)$ cannot linearly read out x on a positive open interval.

3.9. Explore. A partition has three indicator functions whose sets cover the domain without overlap. Explain why appending a constant to all three creates dependence. What happens if one indicator is omitted instead?

3.10. Explore. Specify a dictionary and readout for predicting only the squared first state in Equation (1.1). Then explain why that specification does not solve the two-state recovery task.

Free Sample

3.9 Notes and further reading

Observable coordinates and their interpretation are central to Brunton et al. (2016); Budišić et al. (2012). The examples here separate information, readout, and evolution before introducing fitting. Their usefulness should be judged for the stated output and domain, rather than by dictionary size alone.

Free Sample

Free Sample

Full book available in *Sapientia Scientifica*.

Free Sample